On Efficiently Updating Singular Value Decomposition Based Reduced Order Models

Ralf Zimmermann

GAMM Workshop Applied and Numerical Linear Algebra

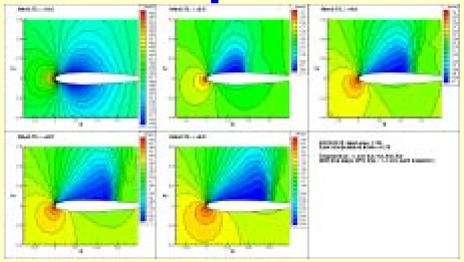
with Special Emphasis on Model Reduction

Bremen, Sep. 22.-23. 2011

DLR Deutsches Zentrum für Luft- und Raumfahrt e.V. in der Helmholtz-Gemeinschaft

The POD-based ROM approach

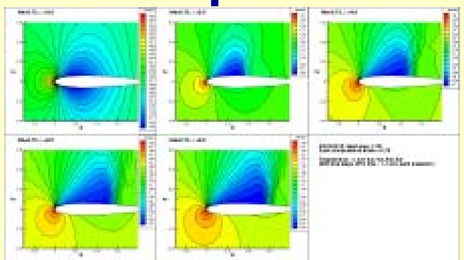
1. Input: CFD snaphots



Flow solutions at m different flow conditions

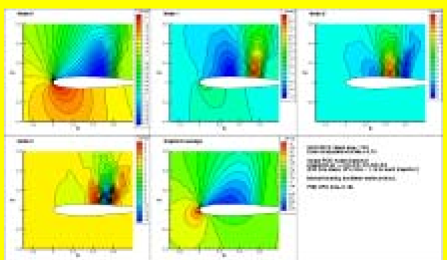
The POD-based ROM approach

1. Input: CFD snaphots



Flow solutions at m different flow conditions

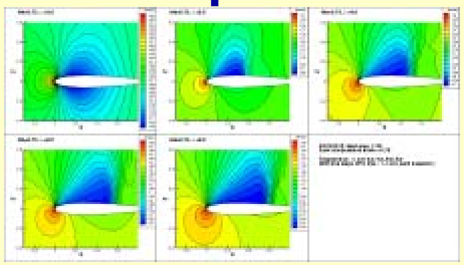
2. POD Basis



Orthonomal basis ordered by infor-mation content spanning the same space

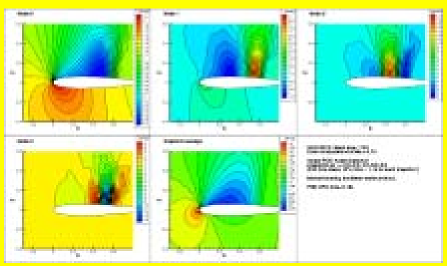
The POD-based ROM approach

1. Input: CFD snaphots



Flow solutions at m different flow conditions

2. POD Basis



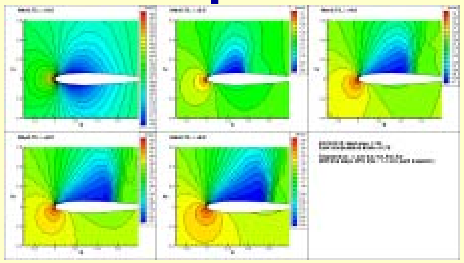
Orthonomal basis ordered by infor-mation content spanning the same space

3. Order Reduction

Select $\tilde{m} < m$ POD components with largest information content

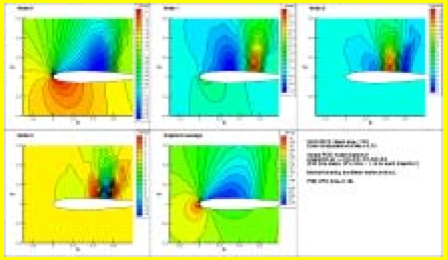
The POD-based ROM approach

1. Input: CFD snaphots



Flow solutions at m different flow conditions

2. POD Basis



Orthonomal basis ordered by information content spanning the same space

3. Order Reduction

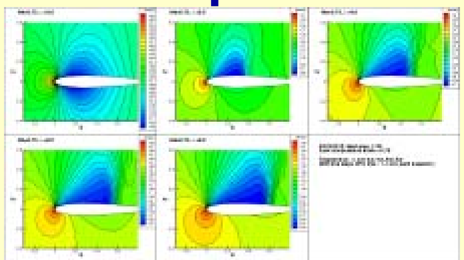
Select $\tilde{m} < m$ POD components with largest information content

4. Prediction step

Determine POD-ROM coefficients by interpolation / solving low-order PDEs / least-squares optimization

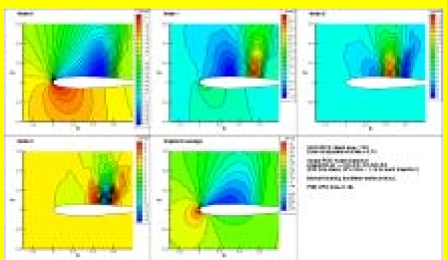
The POD-based ROM approach

1. Input: CFD snaphots



Flow solutions at m different flow conditions

2. POD Basis



Orthonomal basis ordered by infor-mation content spanning the same space

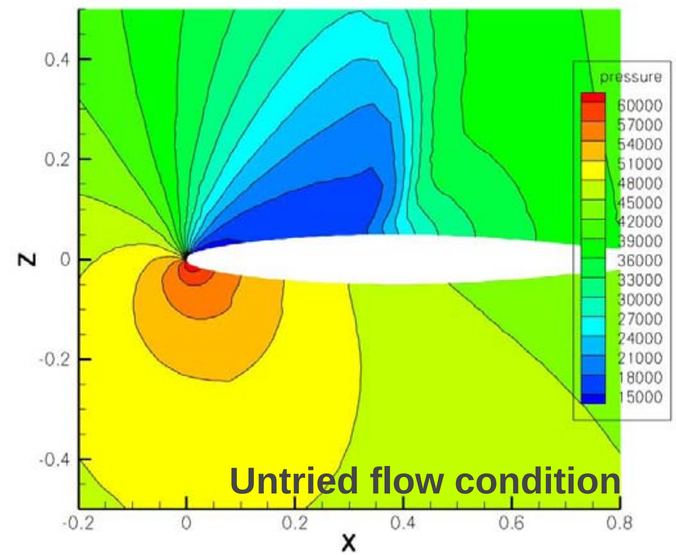
3. Order Reduction

Select $\tilde{m} < m$ POD components with largest information content

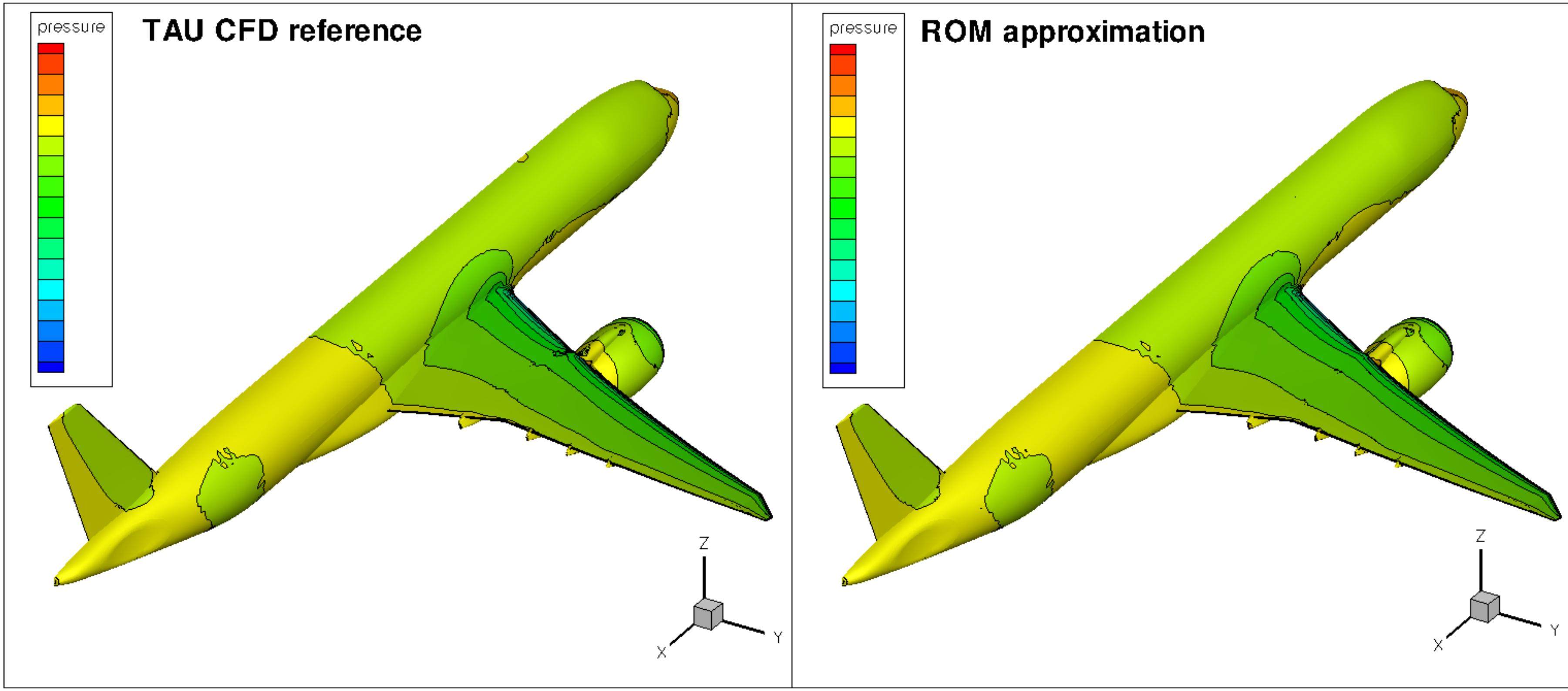
4. Prediction step

Determine POD-ROM coefficients by interpolation / solving low-order PDEs / least-squares optimization

5. Output: approximated flow field



POD based reduced order models are a powerful tool…



Industrial aircraft configuration

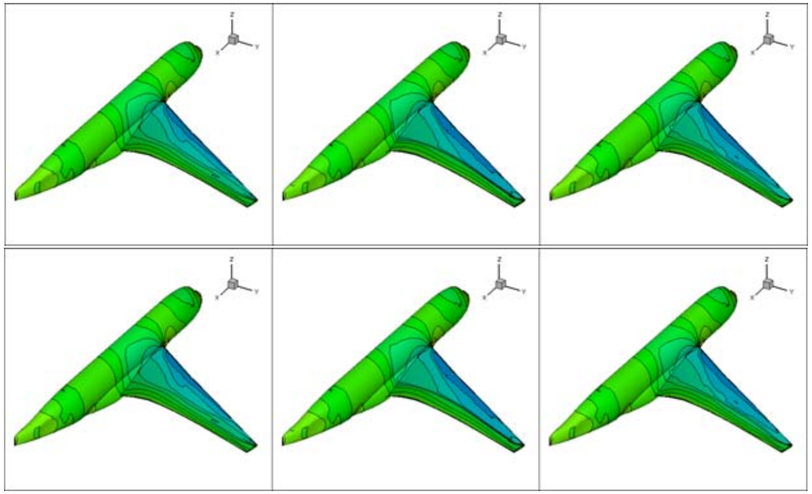
ROM Speed-up factor > 300

Grid size ~**9Mio**, subsonic Ma = 0.2

Snapshot data at AoA = -1°, 0°, 1°, 2°

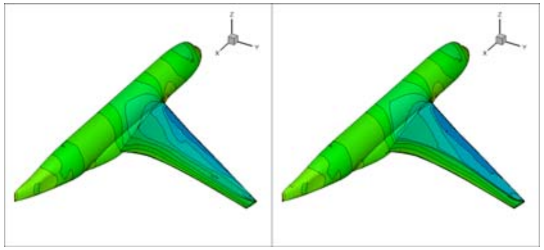
Prediction at AoA = 7° **(Extrapolation!)**

… but treating *large* snapshots may become a challenge!



POD/SVD representation known

+



Incoming new snapshots

How to compute POD/SVD of augmented data set efficiently?

Nomenclature

Snapshot matrix: $Y_m = (W^1, \ldots, W^m) \in R^{n \times m}$

Snapshot average: $A_m = \frac{1}{m} \sum_{j=1}^{m} W^j$

Centering: $\bar{Y}_m = Y_m - A_m 1^T = (W^1 - A_m, \ldots, W^m - A_m) =: (\bar{W}^1, \ldots, \bar{W}^m),$

SVD: $\bar{Y}_m = U \Sigma V^T$

Relative Information content: $ric(r_m) = \frac{\sum_{i=1}^{r_m} \sigma_i^2}{\sum_{i=1}^{m} \sigma_i^2}, \quad r_m \leq m-1$

Reduced-order representation

Discard columns corresponding to small singular values:

$$\bar{Y}_m \approx U_m \Sigma_m V_m^T,$$

$$U_m = \left(U^1,\ldots,U^{r_m}\right) \in R^{n\times r_m}, \quad V_m = \left(V^1,\ldots,V^{r_m}\right) \in R^{m\times r_m}, \quad \Sigma_m = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{r_m} \end{pmatrix}$$

Definition:

The data set $\left(U_m, \Sigma_m, V_m, A_m, n, m, r_m\right)$

is called reduced-order model of order r_m of Y_m.

The ratio $\frac{r_m}{m}$ is called the compression rate.

The SVD basis update problem

Given: ROM $\left(U_m, \Sigma_m, V_m, A_m, n, m, r_m\right)$ of $Y_m \in R^{n \times m}$.

p new snapshot observations $\left(W^{m+1}, \ldots, W^{m+p}\right)$

Task: Compute ROM $\left(U_{m+p}, \Sigma_{m+p}, V_{m+p}, A_{m+p}, n, m+p, r_{m+p}\right)$ of Y_{m+p}.

Requirement:

- use only the previous stage ROM and the incoming snapshots!
- $n >> m$! Keep computational costs depending on n as low as possible

Objective

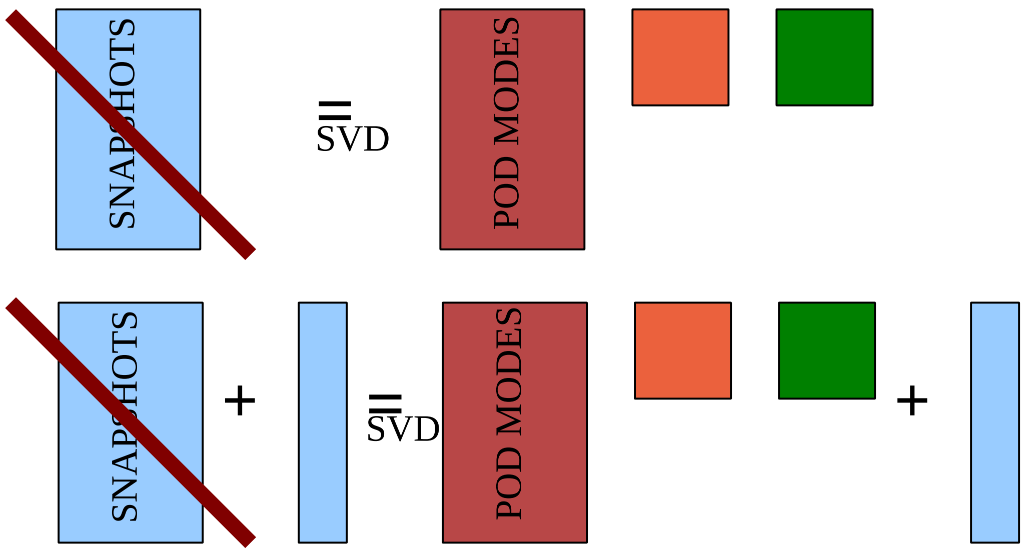
Efficient SVD basis update

- Update SVD basis without having to store the **initial** snapshots

Objective

Efficient SVD basis update

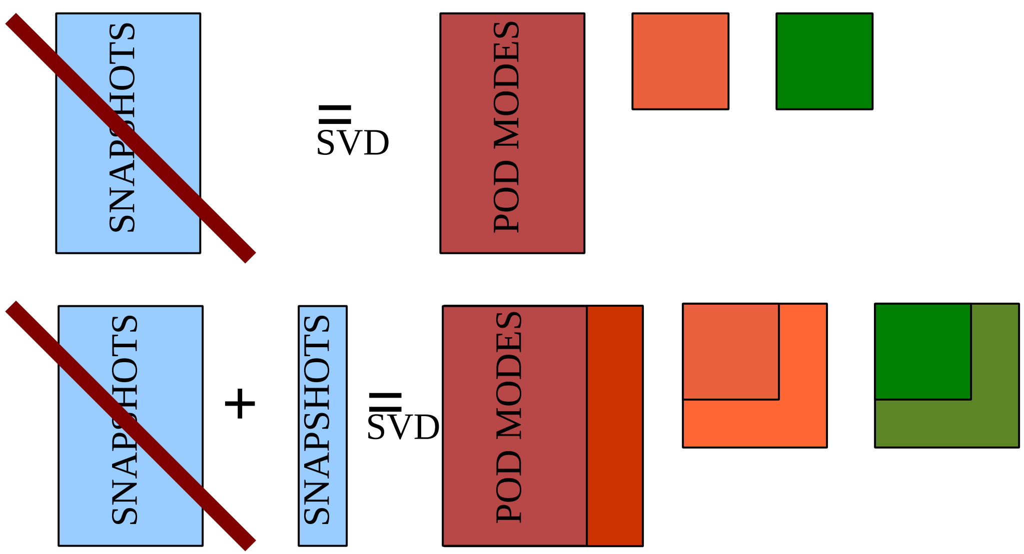
- Update SVD basis without having to store the **initial** snapshots



Objective

Efficient SVD basis update

- Update SVD basis without having to store the **initial** snapshots



SVD basis update strategies

Updating the snapshot mean

Shift vector: $$T_{m+p} := \frac{1}{m+p}\left(pA_m - \sum_{i=m+1}^{m+p} W^i \right)$$

$$\Rightarrow A_{m+p} = A_m - T_{m+p}$$

Shift update snapshots to the previous-stage center:

$$\overline{W}^{m+i} := W^{m+i} - A_m, \quad i = 1, \ldots, p$$

To do: Decompose

$$\overline{Y}_{m+p} = \left(\overline{Y}_m, \overline{W}\right) + T_{m+p} 1^T_{m+p} \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V^T_{m+p},$$

SVD basis update strategies

Updating the snapshot mean

Shift vector: $$T_{m+p} := \frac{1}{m+p}\left(pA_m - \sum_{i=m+1}^{m+p} W^i \right)$$

$$\Rightarrow A_{m+p} = A_m - T_{m+p}$$

Shift update snapshots to the previous-stage center:

$$\overline{W}^{m+i} := W^{m+i} - A_m, \quad i = 1, \ldots, p$$

start here

To do: Decompose

$$\overline{Y}_{m+p} = \left(\overline{Y}_m, \overline{W}\right) + T_{m+p} 1_{m+p}^T \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V_{m+p}^T,$$

SVD basis update strategies (A): two-steps SVD[1,2]

Setting $$X = \left(\overline{Y}_m, 0^{n \times p}\right), \; B^T = \left(0^{p \times m}, I^{p \times p}\right)$$

it holds

$$\left(\overline{Y}_m, \overline{W}\right) = X + \overline{W} B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left(V_m^T, 0^{r_m \times p}\right)$$

[1] M. Brand: "Fast low-rank modifications of the thin SVD", *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: "Merging and splitting eigenspace models", *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (A): two-steps SVD[1,2]

Setting $X = \left(\overline{Y}_m, 0^{n\times p}\right),\ B^T = \left(0^{p\times m}, I^{p\times p}\right)$

it holds

$$\left(\overline{Y}_m, \overline{W}\right) = X + \overline{W}B^T \text{ and } X \approx U_m \Sigma_m \left(V_m^T, 0^{r_m \times p}\right)$$

rank-p SVD update problem

[1] M. Brand: "Fast low-rank modifications of the thin SVD", *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: "Merging and splitting eigenspace models", *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (A): two-steps SVD[1,2]

Setting $X = \left(\overline{Y}_m, 0^{n\times p}\right),\ B^T = \left(0^{p\times m}, I^{p\times p}\right)$

it holds

$$\left(\overline{Y}_m, \overline{W}\right) = X + \overline{W}B^T \ \text{ and } \ X \approx U_m \Sigma_m \left(V_m^T, 0^{r_m \times p}\right)$$

Factoring out orthogonal components:

$$X + \overline{W}B^T = \left(U_m, \overline{W}\right)\begin{pmatrix} \Sigma_m & 0 \\ 0 & I^{p\times p} \end{pmatrix}\begin{pmatrix} (V_m^T, 0^{r_m\times p}) \\ (0^{p\times m}, I^{p\times p}) \end{pmatrix}$$

[1] M. Brand: "Fast low-rank modifications of the thin SVD", *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: "Merging and splitting eigenspace models", *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (A): two-steps SVD[1,2]

Setting $X = \left(\overline{Y}_m, 0^{n\times p}\right), \; B^T = \left(0^{p\times m}, I^{p\times p}\right)$

it holds

$$\left(\overline{Y}_m, \overline{W}\right) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left(V_m^T, 0^{r_m \times p}\right)$$

Factoring out orthogonal components:

$$X + \overline{W}B^T = \left(U_m, P_{orth}\right)\begin{pmatrix} \Sigma_m & U_m^T \overline{W} \\ 0 & P_{orth}^T P \end{pmatrix}\begin{pmatrix} (V_m^T, 0^{r_m\times p}) \\ (0^{p\times m}, I^{p\times p}) \end{pmatrix}$$

where $P = \overline{W} - U(U_m^T \overline{W}), \quad P_{orth} = \operatorname{orth}(P)$ **e.g. via Gram Schmidt**

[1] M. Brand: “Fast low-rank modifications of the thin SVD”, *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: “Merging and splitting eigenspace models”, *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (A): two-steps SVD[1,2]

Setting $X = \left(\overline{Y}_m, 0^{n\times p}\right),\ B^T = \left(0^{p\times m}, I^{p\times p}\right)$

it holds

$$\left(\overline{Y}_m, \overline{W}\right) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left(V_m^T, 0^{r_m \times p}\right)$$

Factoring out orthogonal components:

$$X + \overline{W}B^T = \left(U_m, P_{orth}\right)\begin{pmatrix} \Sigma_m & U_m^T\overline{W} \\ 0 & P_{orth}^T P \end{pmatrix}\begin{pmatrix} (V_m^T, 0^{r_m\times p}) \\ (0^{p\times m}, I^{p\times p}) \end{pmatrix}$$

orthonormal columns

[1] M. Brand: “Fast low-rank modifications of the thin SVD”, *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: “Merging and splitting eigenspace models”, *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (A): two-steps SVD[1,2]

Setting $X = \left(\overline{Y}_m, 0^{n\times p}\right),\ B^T = \left(0^{p\times m}, I^{p\times p}\right)$

it holds

$$\left(\overline{Y}_m, \overline{W}\right) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left(V_m^T, 0^{r_m\times p}\right)$$

Factoring out orthogonal components:

$$X + \overline{W}B^T = \left(U_m, P_{orth}\right)\begin{pmatrix} \Sigma_m & U_m^T\overline{W} \\ 0 & P_{orth}^T P \end{pmatrix}\begin{pmatrix} (V_m^T, 0^{r_m\times p}) \\ (0^{p\times m}, I^{p\times p}) \end{pmatrix}$$

orthonormal columns

[1] M. Brand: "Fast low-rank modifications of the thin SVD", *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: "Merging and splitting eigenspace models", *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (A): two-steps SVD[1,2]

Setting $X = \left(\overline{Y}_m, 0^{n\times p}\right),\; B^T = \left(0^{p\times m}, I^{p\times p}\right)$

it holds

$$\left(\overline{Y}_m, \overline{W}\right) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left(V_m^T, 0^{r_m\times p}\right)$$

Factoring out orthogonal components:

$$X + \overline{W}B^T = \left(U_m, P_{orth}\right)\begin{pmatrix} \Sigma_m & U_m^T\overline{W} \\ 0 & P_{orth}^T P \end{pmatrix}\begin{pmatrix} (V_m^T, 0^{r_m\times p}) \\ (0^{p\times m}, I^{p\times p}) \end{pmatrix}$$

$\boxed{\textbf{size } (r_m + p)\times(r_m + p)}$

[1] M. Brand: “Fast low-rank modifications of the thin SVD”, *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: “Merging and splitting eigenspace models”, *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (A): two-steps SVD[1,2]

Setting $$X = \left(\overline{Y}_m, 0^{n\times p}\right),\; B^T = \left(0^{p\times m}, I^{p\times p}\right)$$

it holds

$$\left(\overline{Y}_m, \overline{W}\right) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left(V_m^T, 0^{r_m\times p}\right)$$

Factoring out orthogonal components:

$$X + \overline{W}B^T = \left(U_m, P_{orth}\right)\left(\tilde{U}\,\tilde{\Sigma}\,\tilde{V}^T\right)\begin{pmatrix} (V_m^T, 0^{r_m\times p}) \\ (0^{p\times m}, I^{p\times p}) \end{pmatrix}$$

[1] M. Brand: “Fast low-rank modifications of the thin SVD”, *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: “Merging and splitting eigenspace models”, *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (A): two-steps SVD[1,2]

Setting $X = \left(\overline{Y}_m, 0^{n\times p}\right),\ B^T = \left(0^{p\times m}, I^{p\times p}\right)$

it holds

$$\left(\overline{Y}_m, \overline{W}\right) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left(V_m^T, 0^{r_m \times p}\right)$$

Factoring out orthogonal components:

$$X + \overline{W}B^T = \left[\left(U_m, P_{orth}\right)\tilde{U}\right] \tilde{\Sigma} \left[\tilde{V}^T \begin{pmatrix} (V_m^T, 0^{r_m\times p}) \\ (0^{p\times m}, I^{p\times p}) \end{pmatrix}\right]$$

[1] M. Brand: “Fast low-rank modifications of the thin SVD”, *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: “Merging and splitting eigenspace models”, *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (A): two-steps SVD[1,2]

Repeat for shifting to new center:

$$\bar{Y}_{m+p} = \left(\bar{Y}_m, \bar{W}\right) + T_{m+p} 1^T_{m+p} \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V^T_{m+p},$$

SVD now known

[1] M. Brand: “Fast low-rank modifications of the thin SVD”, *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: “Merging and splitting eigenspace models”, *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (A): two-steps SVD[1,2]

Comments:

- 're-orthogonalization' expensive, additional (large-scale) SVD required
- less robust via Gram-Schmidt

$$P = \overline{W} - U(U_m^T \overline{W}), \quad P_{orth} = \mathrm{orth}(P)$$

- Parallelization is more involved

[1] M. Brand: "Fast low-rank modifications of the thin SVD", *Lin. Alg. and its Appl.* 415, 2006

[2] P. Hall et al.: "Merging and splitting eigenspace models", *IEEE Trans. Pattern analysis and Machine Intelligence*, 22(9), 2000

SVD basis update strategies (B): EVD[3]+SVD

Objective: $$\overline{Y}_{m+p} = \left(\overline{Y}_m, \overline{W}\right) + T_{m+p} 1_{m+p}^T \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V_{m+p}^T,$$

First step: Compute SVD of $\left(\overline{Y}_m, \overline{W}\right)$

Reduce to symmetric EVD

$$\left(\overline{Y}_m, \overline{W}\right)^T \left(\overline{Y}_m, \overline{W}\right) = \begin{pmatrix} \overline{Y}_m{}^T \overline{Y}_m & \overline{Y}_m{}^T \overline{W} \\ \overline{W}^T \overline{Y}_m & \overline{W}^T \overline{W} \end{pmatrix}$$

[3] Generalize ideas from: J.R. Bunch, C.P. Nielsen: “Updating the singular value decomposition”, N*umer. Math. 31, 1978*

SVD basis update strategies (B): EVD[3]+SVD

Objective: $$\overline{Y}_{m+p} = \left(\overline{Y}_m, \overline{W}\right) + T_{m+p} 1_{m+p}^T \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V_{m+p}^T,$$

First step: Compute SVD of $\left(\overline{Y}_m, \overline{W}\right)$

Reduce to symmetric EVD

$$\left(\overline{Y}_m, \overline{W}\right)^T \left(\overline{Y}_m, \overline{W}\right) \approx \begin{pmatrix} V_m \Sigma_m^2 V_m^T & V_m \Sigma_m U_m^T \overline{W} \\ \overline{W}^T U_m \Sigma_m V_m^T & \overline{W}^T \overline{W} \end{pmatrix}$$

Exploit previous stage SVD

[3] Generalize ideas from: J.R. Bunch, C.P. Nielsen: “Updating the singular value decomposition”, N*umer. Math. 31, 1978*

SVD basis update strategies (B): EVD³+SVD

Objective: $\bar{Y}_{m+p} = \left(\bar{Y}_m, \bar{W}\right) + T_{m+p} 1_{m+p}^T \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V_{m+p}^T,$

First step: Compute SVD of $\left(\bar{Y}_m, \bar{W}\right)$

Reduce to symmetric EVD

$$\left(\bar{Y}_m, \bar{W}\right)^T \left(\bar{Y}_m, \bar{W}\right) \approx \begin{pmatrix} V_m & 0 \\ 0 & I^{p\times p} \end{pmatrix} \begin{pmatrix} \Sigma_m^2 & \Sigma_m U_m^T \bar{W} \\ \bar{W}^T U_m \Sigma_m & \bar{W}^T \bar{W} \end{pmatrix} \begin{pmatrix} V_m^T & 0 \\ 0 & I^{p\times p} \end{pmatrix}$$

factor out

[3] Generalize ideas from: J.R. Bunch, C.P. Nielsen: "Updating the singular value decomposition", N*umer. Math. 31, 1978*

SVD basis update strategies (B): EVD[3]+SVD

Objective: $$\bar{Y}_{m+p} = \left(\bar{Y}_m, \bar{W}\right) + T_{m+p} 1_{m+p}^T \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V_{m+p}^T,$$

First step: Compute SVD of $\left(\bar{Y}_m, \bar{W}\right)$

Reduce to symmetric EVD

$$\left(\bar{Y}_m, \bar{W}\right)^T \left(\bar{Y}_m, \bar{W}\right) \approx \begin{pmatrix} V_m & 0 \\ 0 & I^{p\times p} \end{pmatrix} \begin{pmatrix} \Sigma_m^2 & \Sigma_m U_m^T \bar{W} \\ \bar{W}^T U_m \Sigma_m & \bar{W}^T \bar{W} \end{pmatrix} \begin{pmatrix} V_m^T & 0 \\ 0 & I^{p\times p} \end{pmatrix}$$

size $(r_m + p) \times (r_m + p)$

[3] Generalize ideas from: J.R. Bunch, C.P. Nielsen: “Updating the singular value decomposition”, N*umer. Math. 31, 1978*

DLR C²A²S²E

SVD basis update strategies (B): EVD[3]+SVD

Objective: $$\overline{Y}_{m+p} = \left(\overline{Y}_m, \overline{W}\right) + T_{m+p} 1_{m+p}^T \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V_{m+p}^T,$$

First step: Compute SVD of $\left(\overline{Y}_m, \overline{W}\right)$

Reduce to symmetric EVD

$$\left(\overline{Y}_m, \overline{W}\right)^T \left(\overline{Y}_m, \overline{W}\right) \approx \begin{pmatrix} V_m & 0 \\ 0 & I^{p\times p} \end{pmatrix} \left(\tilde{Q}\tilde{\Lambda}\tilde{Q}^T\right) \begin{pmatrix} V_m^T & 0 \\ 0 & I^{p\times p} \end{pmatrix}$$

[3] Generalize ideas from: J.R. Bunch, C.P. Nielsen: “Updating the singular value decomposition”, N*umer. Math. 31, 1978*

SVD basis update strategies (B): EVD[3]+SVD

Objective: $\overline{Y}_{m+p} = \left(\overline{Y}_m, \overline{W}\right) + T_{m+p} 1_{m+p}^T \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V_{m+p}^T,$

First step: Compute SVD of $\left(\overline{Y}_m, \overline{W}\right)$

Reduce to symmetric EVD

$$\left(\overline{Y}_m, \overline{W}\right)^T \left(\overline{Y}_m, \overline{W}\right) \approx \begin{pmatrix} V_m & 0 \\ 0 & I^{p\times p} \end{pmatrix} \left(\tilde{Q}\tilde{\Lambda}\tilde{Q}^T\right) \begin{pmatrix} V_m^T & 0 \\ 0 & I^{p\times p} \end{pmatrix}$$

Compute left singular vectors via

$$\hat{U} = \left(U_m \Sigma_m, \overline{W}\right) \tilde{Q} \sqrt{\tilde{\Lambda}^{-1}}$$

[3] Generalize ideas from: J.R. Bunch, C.P. Nielsen: “Updating the singular value decomposition”, N*umer. Math. 31, 1978*

SVD basis update strategies (B): EVD[3] + SVD

Use Brand's method for shifting to new center:

$$\bar{Y}_{m+p} = \left(\bar{Y}_m, \bar{W}\right) + T_{m+p} 1^T_{m+p} \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V^T_{m+p},$$

SVD now known

[3] Generalize ideas from: J.R. Bunch, C.P. Nielsen: "Updating the singular value decomposition", N*umer. Math. 31, 1978*

SVD basis update strategies (C): one-step EVD

Objective: $$\bar{Y}_{m+p} = \left(\bar{Y}_m, \bar{W}\right) + T_{m+p} 1_{m+p}^T \overset{!}{\approx} U_{m+p} \Sigma_{m+p} V_{m+p}^T,$$

Let $X := \left(\bar{Y}_m, \bar{W}\right) + T_{m+p} 1_{m+p}^T \in R^{n\times(m+p)}$

Reduce to symmetric EVD,

exploit previous stage SVD in matrix products

$$X^T X = \tilde{Q}\tilde{\Lambda}\tilde{Q}^T \in R^{(m+p)\times(m+p)}$$

Compute left singular vectors via

$$U_{m+p} = X\tilde{Q}\sqrt{\tilde{\Lambda}^{-1}}$$

SVD basis update strategies (B) and (C)

Comments:

- Straight forward parallelization!

  (only standard matrix products required in parallel)

Analysis of Computational costs

- Count nmp flops for matrix product $AB, \quad A \in R^{n \times m}, B \in R^{m \times p}$

- **Strategy (B) is more efficient than strategy (A):**

$$\mathrm{flops(A)} - \mathrm{flops(B)} = \left(r_m p + \tfrac{1}{2}(p^2 + p)\right) n + O(orth(P))$$
$$= O((mp + p^2)n)$$

Analysis of Computational costs

The ranking of Strategy (B) vs. (C) depends on the compression rate!

- Assumption:

$$r_{m+p} = r_m + p = xm + p, \quad x = \frac{r_m}{m} \text{ (compression rate)}$$

- Then the computational costs differ by

$$\text{flops(C)} - \text{flops(B)} = n\left((x - 2x^2)m^2 + (2 - 3x)(mp + m) - p^2 - p - 2\right)$$

Analysis of Computational costs

The ranking of Strategy (B) vs. (C) depends on the compression rate!

- Solving the quadratic equation shows that
  Strategy (B) is more efficient than Strategy (C) if

$$0 < x < \tfrac{1}{4m}\left(m - 3(p+1) + \sqrt{10m(p+1) + m^2 + p^2 + 10p - 7}\right)$$

- In practical implementations, use switch to select the most efficient update strategy.

Analysis of Computational costs

Rules of thumb:

- **For highly compressed models, Strategy (B) is more efficient than Strategy (C)**
- **For weakly compressed models, the opposite holds true**

More precisely

- Strategy (B) is more efficient than Strategy (C), if $x \leq \frac{1}{2}$ and $m \geq 2p+1$.

- Strategy (C) is more efficient than Strategy (B), if $x \geq \frac{2}{3}$.

Example: Updating SVDs of Random matrices

$$Y \in R^{n\times m}, W \in R^{n\times p}, n = 100{,}000, m = 500, \; p = 100$$

Method	Reconstruction error	time (sec)
Strategy (A), SVD-orth	2.504e-12	7.88
Strategy (A), QR-orth	2.333e-12	8.15
Strategy (B)	2.342e-12	6.32
Strategy (C)	2.279e-12	**4.42**

- Uncompressed models $r_m = m - 1 = 499, \quad r_{m+p} = 599$

Example: Updating SVDs of Random matrices

$$Y \in R^{n\times m}, W \in R^{n\times p}, n = 100{,}000, m = 500, \ p = 100$$

Method	Reconstruction error	time (sec)
Strategy (A), SVD-orth	92.21	4.18
Strategy (A), QR-orth	92.21	4.23
Strategy (B)	92.21	**1.93**
Strategy (C)	93.41	3.54

- Compression level $r_m = 200, \quad r_{m+p} = 200, \quad x = \frac{r_m}{m} = 0.4$

DLR C²A²S²E

Summary

- The update strategies following the symmetric EVD approach are more efficient than the SVD update known from literature (Brand, Hall et al.)

  (Assumption: n>>m)

- Industrial point of view: SVD/EVD performed by **'black box function'**

- Most efficient: choose method depending on the compression rate

- The examples suggest that all methods share a similar level of accuracy

  (SVD-approach may suffer from orthogonal out-factoring,

  EVD-approaches may suffer from squaring the condition number)

Summary

- For details and additional references, see:

  "A comprehensive comparison of various algorithms for efficiently updating singular value decomposition based reduced order models", DLR IB 124-2011/3

Freely available at DLR's electronic library:

http://elib.dlr.de/70251

Thank you for your attention!